

DOCKET No.

BEVOP024

U.S. PATENT APPLICATION  
FOR  
SYSTEM, METHOD AND COMPUTER PROGRAM  
PRODUCT FOR EXTENDED ELEMENT TYPES TO  
ENHANCE OPERATIONAL CHARACTERISTICS  
IN A VOICE PORTAL

ASSIGNEE: BEVOCAL, INC.

KEVIN J. ZILKA

P.O. Box 721120

SAN JOSE, CA 95172

# SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR EXTENDED ELEMENT TYPES TO ENHANCE OPERATIONAL CHARACTERISTICS IN A VOICE PORTAL

5

## FIELD OF THE INVENTION

The present invention relates to voice portals, and more particularly to the use of a voice extensible mark-up language in a voice portal.

10

## BACKGROUND OF THE INVENTION

Techniques for accomplishing automatic speech recognition (ASR) are well known. Among known ASR techniques are those that use grammars. A grammar is a representation of the language or phrases expected to be used or spoken in a given context. In one sense, then, ASR grammars typically constrain the speech recognizer to a vocabulary that is a subset of the universe of potentially-spoken words; and grammars may include subgrammars. An ASR grammar rule can then be used to represent the set of "phrases" or combinations of words from one or more grammars or subgrammars that may be expected in a given context. "Grammar" may also refer generally to a statistical language model (where a model represents phrases), such as those used in language understanding systems.

Products and services that utilize some form of automatic speech recognition ("ASR") methodology have been recently introduced commercially. Desirable attributes of complex ASR services that would utilize such ASR technology include high accuracy in recognition; robustness to enable recognition where speakers have differing accents or dialects, and/or in the presence of background noise; ability to handle large vocabularies; and natural language understanding. In order to achieve these attributes for complex ASR services, ASR techniques and engines typically

require computer-based systems having significant processing capability in order to achieve the desired speech recognition capability.

In a standard speech recognition/synthesis system, a database of utterances is  
5 maintained for administering a predetermined service. In one example of operation, a user may utilize a telecommunication network to communicate utterances to the system. In response to such communication, the utterances are recognized utilizing speech recognition, and processing takes place utilizing the recognized utterances. Thereafter, synthesized speech is outputted in accordance with the processing. In  
10 one particular application, a user may verbally communicate a street address to the speech recognition system, and driving directions may be returned utilizing synthesized speech.

In order to facilitate the interaction between the user and a system that is available  
15 through the Internet, a specially adapted voice mark-up language (VoiceXML) is employed. VoiceXML allows for the creation of voice dialogs, which are stored on any Web site and referenced by URL just like HTML documents. In use, the user may call a phone number and interact with a VoiceXML application through speech recognition, and (TTS) Text-To-Speech and recorded prompts. To accomplish this,  
20 VoiceXML allows a developer to create a script, whereby the user can have a conversation with a script which is stored on the Web site, and executed by a VoiceXML Browser. The user places a call and is connected to a program called a voice browser, or "interpreter". The voice browser will fetch the user's VoiceXML document at a specified URL. The user will interact with the VoiceXML document  
25 using speech recognition as it is interpreted by the VoiceXML Browser. The markup defined in VoiceXML is a specific instance of the Extensible Markup Language (XML), the strategic data definition language for the Internet.

VoiceXML offers a standard format in which developers can create voice dialog  
30 with a Web site. Unfortunately, such standard is often limiting in the functionality

that it provides. As such, there is thus a need for extending VoiceXML functionality in the context of a speech recognition/synthesis system.

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
22

### DISCLOSURE OF THE INVENTION

A system, method and computer program product are provided for dynamically extending element types for a voice-based extensible mark-up language

5 (VoiceXML). Initially, a plurality of element types are registered with a VoiceXML interpreter. In use, such registered element types are received during use of the VoiceXML interpreter. In response to such receipt, code associated with the registered element types is accessed utilizing the VoiceXML interpreter. Such code extends the functionality of the VoiceXML.

10

In one embodiment, the code is written in JAVA. Of course, other computer languages may be used per the desires of the developer.

15

In another embodiment, the registration may include tagging the registered element types as being extensions to a conventional set of element types. Further, the element types may be tagged utilizing extensible mark-up language (XML) namespaces. Still yet, the registration may further include identifying a VoiceXML element type to be extended, along with a name for the to-be-extended VoiceXML element type.

20

Thereafter, the registration includes identifying a class to be loaded for the VoiceXML element type to be extended, and a location of a file containing class files associated with the identified class.

25

A system, method and computer program product are also provided for dynamically extending a type attribute of elements of a voice-based extensible mark-up language (VoiceXML). Initially, an extended type attribute associated with an element of VoiceXML is registered with a VoiceXML interpreter. During use, the element may be received, and the extended type attribute associated with the element is identified. Thereafter, code corresponding to the registered type attribute may be accessed utilizing the VoiceXML interpreter. Such code extends the functionality of the

30 VoiceXML.

As such, a data structure is provided for dynamically extending a type attribute of elements of a VoiceXML. First provided is a VoiceXML type attribute object that is extended to include a previously undefined type attribute. Also included is a

- 5 VoiceXML element. Associated therewith is a class object for identifying a class to be loaded for the VoiceXML type attribute object that is extended. In use, the data structure is capable of being used to register type attributes capable of accessing code to extend the functionality of the VoiceXML.
- 10 The present embodiment thus provides a basic technique of modifying or adding to a mapping between VoiceXML tag names and Java classes which implement those tags. The interpreter looks at this mapping to discover which classes should be used to implement a specific tag.
- 15 The present embodiment further provides a syntax by which this extension of tags may be accomplished utilizing VoiceXML. This allows a VoiceXML developer to specify the extension and classes which should be used.

**BRIEF DESCRIPTION OF THE DRAWINGS**

5     Figure 1 illustrates an exemplary environment in which the present invention may be implemented;

Figure 2 shows a representative hardware environment associated with the various components of Figure 1;

10

Figure 3 illustrates a method for providing a speech recognition process utilizing the utterances collected during the method of Figure 3;

15     Figure 4 illustrates a web-based interface which interacts with a database to enable and coordinate an audio transcription effort;

Figure 5 is a schematic illustrating the manner in which VoiceXML functions, in accordance with one embodiment of the present invention; and

20     Figure 6 illustrates a method of dynamically extending element types for VoiceXML.

25

### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates one exemplary platform **150** on which the present invention may  
5 be implemented. The present platform **150** is capable of supporting voice  
applications that provide unique business services. Such voice applications may be  
adapted for consumer services or internal applications for employee productivity.

The present platform of Figure 1 provides an end-to-end solution that manages a  
10 presentation layer **152**, application logic **154**, information access services **156**, and  
telecom infrastructure **159**. With the instant platform, customers can build complex  
voice applications through a suite of customized applications and a rich development  
tool set on an application server **160**. The present platform **150** is capable of  
15 deploying applications in a reliable, scalable manner, and maintaining the entire  
system through monitoring tools.

The present platform **150** is multi-modal in that it facilitates information delivery via  
multiple mechanisms **162**, i.e. Voice, Wireless Application Protocol (WAP),  
Hypertext Mark-up Language (HTML), Facsimile, Electronic Mail, Pager, and Short  
20 Message Service (SMS). It further includes a VoiceXML interpreter **164** that is fully  
compliant with the VoiceXML 1.0 specification, written entirely in Java®, and  
supports Nuance® SpeechObjects **166**.

Yet another feature of the present platform **150** is its modular architecture, enabling  
25 “plug-and-play” capabilities. Still yet, the instant platform **150** is extensible in that  
developers can create their own custom services to extend the platform **150**. For  
further versatility, Java® based components are supported that enable rapid  
development, reliability, and portability. Another web server **168** supports a web-  
based development environment that provides a comprehensive set of tools and



resources which developers may need to create their own innovative speech applications.

- Support for SIP and SS7 (Signaling System 7) is also provided. Backend Services
- 5   **172** are also included that provide value added functionality such as content management **180** and user profile management **182**. Still yet, there is support for external billing engines **174** and integration of leading edge technologies from Nuance®, Oracle®, Cisco®, Natural Microsystems®, and Sun Microsystems®.
- 10   More information will now be set forth regarding the application layer **154**, presentation layer **152**, and services layer **156**.

#### Application Layer (154)

- 15   The application layer **154** provides a set of reusable application components as well as the software engine for their execution. Through this layer, applications benefit from a reliable, scalable, and high performing operating environment. The application server **160** automatically handles lower level details such as system management, communications, monitoring, scheduling, logging, and load balancing.
- 20   Some optional features associated with each of the various components of the application layer **154** will now be set forth.

#### *Application Server (160)*

- 25       • A high performance web/JSP server that hosts the business and presentation logic of applications.
- High performance, load balanced, with failover.
- Contains reusable application components and ready to use applications.
- Hosts Java Servlets and JSP's for custom applications.
- 30       • Provides easy to use taglib access to platform services.

### *VoiceXML Interpreter (164)*

- Executes VoiceXML applications
- 5 • VoiceXML 1.0 compliant
- Can execute applications hosted on either side of the firewall.
- Extensions for easy access to system services such as billing.
- Extensible - allows installation of custom VoiceXML tag libraries and speech objects.
- 10 • Provides access to SpeechObjects **166** from VoiceXML.
- Integrated with debugging and monitoring tools.
- Written in Java®.

### *Speech Objects Server (166)*

- 15 • Hosts SpeechObjects based components.
- Provides a platform for running SpeechObjects based applications.
- Contains a rich library of reusable SpeechObjects.

### 20 Services Layer (156)

The services layer **156** simplifies the development of voice applications by providing access to modular value-added services. These backend modules deliver a complete set of functionality, and handle low level processing such as error checking.

- 25 Examples of services include the content **180**, user profile **182**, billing **174**, and portal management **184** services. By this design, developers can create high performing, enterprise applications without complex programming. Some optional features associated with each of the various components of the services layer **156** will now be set forth.

### *Content (180)*

- 5
  - Manages content feeds and databases such as weather reports, stock quotes, and sports.
  - Ensures content is received and processed appropriately.
  - Provides content only upon authenticated request.
  - Communicates with logging service **186** to track content usage for auditing purposes.
- 10
  - Supports multiple, redundant content feeds with automatic failover.
  - Sends alarms through alarm service **188**.

### *User Profile (182)*

- 15
  - Manages user database
  - Can connect to a 3<sup>rd</sup> party user database **190**. For example, if a customer wants to leverage his/her own user database, this service will manage the connection to the external user database.
  - Provides user information upon authenticated request.

20

### *Alarm (188)*

- 25
  - Provides a simple, uniform way for system components to report a wide variety of alarms.
  - Allows for notification (Simply Network Management Protocol (SNMP), telephone, electronic mail, pager, facsimile, SMS, WAP push, etc.) based on alarm conditions.
  - Allows for alarm management (assignment, status tracking, etc) and integration with trouble ticketing and/or helpdesk systems.

- Allows for integration of alarms into customer premise environments.

#### *Configuration Management (191)*

- 5
- Maintains the configuration of the entire system.

#### *Performance Monitor (193)*

- 10
- Provides real time monitoring of entire system such as number of simultaneous users per customer, number of users in a given application, and the uptime of the system.
  - Enables customers to determine performance of system at any instance.

#### *Portal Management (184)*

- 15
- The portal management service **184** maintains information on the configuration of each voice portal and enables customers to electronically administer their voice portal through the administration web site.
  - Portals can be highly customized by choosing from multiple applications and voices. For example, a customer can configure different packages of applications i.e. a basic package consisting of 3 applications for \$4.95, a deluxe package consisting of 10 applications for \$9.95, and premium package consisting of any 20 applications for \$14.95.
- 20

#### *Instant Messenger (192)*

- 25
- Detects when users are “on-line” and can pass messages such as new voicemails and e-mails to these users.

### *Billing (174)*

- Provides billing infrastructure such as capturing and processing billable events, rating, and interfaces to external billing systems.

5

### *Logging (186)*

- Logs all events sent over the JMS bus **194**. Examples include User A of Company ABC accessed Stock Quotes, application server **160** requested driving directions from content service **180**, etc.

10

### *Location (196)*

- Provides geographic location of caller.
- Location service sends a request to the wireless carrier or to a location network service provider such as TimesThree® or US Wireless. The network provider responds with the geographic location (accurate within 75 meters) of the cell phone caller.

15

### *Advertising (197)*

- Administers the insertion of advertisements within each call. The advertising service can deliver targeted ads based on user profile information.
- Interfaces to external advertising services such as Wyndwire® are provided.

20

### *Transactions (198)*

- Provides transaction infrastructure such as shopping cart, tax and shipping calculations, and interfaces to external payment systems.

25

### *Notification (199)*

30

- Provides external and internal notifications based on a timer or on external events such as stock price movements. For example, a user can request that he/she receive a telephone call every day at 8AM.
- 5 • Services can request that they receive a notification to perform an action at a pre-determined time. For example, the content service **180** can request that it receive an instruction every night to archive old content.

### *3<sup>rd</sup> Party Service Adapter (190)*

10

- Enables 3<sup>rd</sup> parties to develop and use their own external services. For instance, if a customer wants to leverage a proprietary system, the 3<sup>rd</sup> party service adapter can enable it as a service that is available to applications.

### Presentation Layer (152)

15

The presentation layer **152** provides the mechanism for communicating with the end user. While the application layer **154** manages the application logic, the presentation layer **152** translates the core logic into a medium that a user's device can understand.

- 20 Thus, the presentation layer **152** enables multi-modal support. For instance, end users can interact with the platform through a telephone, WAP session, HTML session, pager, SMS, facsimile, and electronic mail. Furthermore, as new "touchpoints" emerge, additional modules can seamlessly be integrated into the presentation layer **152** to support them.

25

### *Telephony Server (158)*

- The telephony server **158** provides the interface between the telephony world, both Voice over Internet Protocol (VoIP) and Public Switched Telephone Network (PSTN), and the applications running on the platform. It also provides the interface
- 30

to speech recognition and synthesis engines **153**. Through the telephony server **158**, one can interface to other 3<sup>rd</sup> party application servers **190** such as unified messaging and conferencing server. The telephony server **158** connects to the telephony switches and "handles" the phone call.

5

Features of the telephony server **158** include:

- Mission critical reliability.
- Suite of operations and maintenance tools.
- Telephony connectivity via ISDN/T1/E1, SIP and SS7 protocols.
- 10 • DSP-based telephony boards offload the host, providing real-time echo cancellation, DTMF & call progress detection, and audio compression/decompression.

#### *Speech Recognition Server (155)*

15

The speech recognition server **155** performs speech recognition on real time voice streams from the telephony server **158**. The speech recognition server **155** may support the following features:

- 20 • Carrier grade scalability & reliability
- Large vocabulary size
- Industry leading speaker independent recognition accuracy
- Recognition enhancements for wireless and hands free callers
- Dynamic grammar support – grammars can be added during run time.
- 25 • Multi-language support
- Barge in – enables users to interrupt voice applications. For example, if a user hears “Please say a name of a football team that you,” the user can interject by saying “Miami Dolphins” before the system finishes.
- Speech objects provide easy to use reusable components
- 30 • "On the fly" grammar updates

- Speaker verification

#### *Audio Manager (157)*

- 5
- Manages the prompt server, text-to-speech server, and streaming audio.

#### *Prompt Server (153)*

- 10
- The Prompt server is responsible for caching and managing pre-recorded audio files for a pool of telephony servers.

#### *Text-to-Speech Server (153)*

15

When pre-recorded prompts are unavailable, the text-to-speech server is responsible for transforming text input into audio output that can be streamed to callers on the telephony server **158**. The use of the TTS server offloads the telephony server **158** and allows pools of TTS resources to be shared across several telephony servers. Features include:

- 20
- Support for industry leading technologies such as SpeechWorks® Speechify® and L&H RealSpeak®.
  - Standard Application Program Interface (API) for integration of other TTS engines.

#### *Streaming Audio*

25

The streaming audio server enables static and dynamic audio files to be played to the caller. For instance, a one minute audio news feed would be handled by the streaming audio server.

- 30
- Support for standard static file formats such as WAV and MP3



- Support for streaming (dynamic) file formats such as Real Audio® and Windows® Media®.

#### *PSTN Connectivity*

5

- Support for standard telephony protocols like ISDN, E&M WinkStart®, and various flavors of E1 allow the telephony server **158** to connect to a PBX or local central office.

#### *SIP Connectivity*

10

The platform supports telephony signaling via the Session Initiation Protocol (SIP). The SIP signaling is independent of the audio stream, which is typically provided as a G.711 RTP stream. The use of a SIP enabled network can be used to provide many powerful features including:

15

- Flexible call routing
- Call forwarding
- Blind & supervised transfers
- Location/presence services
- Interoperable with SIP compliant devices such as soft switches
- Direct connectivity to SIP enabled carriers and networks
- Connection to SS7 and standard telephony networks (via gateways)

20

#### *Admin Web Server*

25

- Serves as the primary interface for customers.
- Enables portal management services and provides billing and simple reporting information. It also permits customers to enter problem ticket

orders, modify application content such as advertisements, and perform other value added functions.

- Consists of a website with backend logic tied to the services and application layers. Access to the site is limited to those with a valid user id and password and to those coming from a registered IP address. Once logged in, customers are presented with a homepage that provides access to all available customer resources.

#### *Other (168)*

10

Web-based development environment that provides all the tools and resources developers need to create their own speech applications.

Provides a VoiceXML Interpreter that is:

15

- Compliant with the VoiceXML 1.0 specification.
- Compatible with compelling, location-relevant SpeechObjects -- including grammars for nationwide US street addresses.
- Provides unique tools that are critical to speech application development such as a vocal player. The vocal player addresses usability testing by giving developers convenient access to audio files of real user interactions with their speech applications. This provides an invaluable feedback loop for improving dialogue design.

20

#### *25 WAP, HTML, SMS, Email, Pager, and Fax Gateways*

- Provide access to external browsing devices.
- Manage (establish, maintain, and terminate) connections to external browsing and output devices.
- Encapsulate the details of communicating with external device.

30

- Support both input and output on media where appropriate. For instance, both input from and output to WAP devices.
- Reliably deliver content and notifications.

5 Figure 2 shows a representative hardware environment associated with the various systems, i.e. computers, servers, etc., of Figure 1. Figure 2 illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **210**, such as a microprocessor, and a number of other units interconnected via a system bus **212**.

10

The workstation shown in Figure 2 includes a Random Access Memory (RAM) **214**, Read Only Memory (ROM) **216**, an I/O adapter **218** for connecting peripheral devices such as disk storage units **220** to the bus **212**, a user interface adapter **222** for connecting a keyboard **224**, a mouse **226**, a speaker **228**, a microphone **232**, and/or  
15 other user interface devices such as a touch screen (not shown) to the bus **212**, communication adapter **234** for connecting the workstation to a communication network (e.g., a data processing network) and a display adapter **236** for connecting the bus **212** to a display device **238**. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating  
20 System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

Figure 3 illustrates a method **350** for providing a speech recognition process utilizing  
25 the utterances collected during use of a voice portal. Initially, a database of the collected utterances is maintained. See operation **352**. In operation **354**, information associated with the utterances is collected utilizing a speech recognition process. When a speech recognition process application is deployed, audio data and recognition logs may be created. Such data and logs may also be created by simply  
30 parsing through the database at any desired time.

In one embodiment, a database record may be created for each utterance. Table 1 illustrates the various information that the record may include.

5

Table 1

- Name of the grammar it was recognized against;
- Name of the audio file on disk;
- Directory path to that audio file;
- 10 • Size of the file (which in turn can be used to calculate the length of the utterance if the sampling rate is fixed);
- Session identifier;
- Index of the utterance (i.e. the number of utterances said before in the same session);
- 15 • Dialog state (identifier indicating context in the dialog flow in which recognition happened);
- Recognition status (i.e. what the recognizer did with the utterance (rejected, recognized, recognizer was too slow);
- Recognition confidence associated with the recognition result;
- 20 • Recognition hypothesis;
- Gender of the speaker;
- Identification of the transcriber; and/or
- Date the utterances were transcribed.

25 Inserting utterances and associated information in this fashion in the database (SQL database) allows instant visibility into the data collected. Table 2 illustrates the variety of information that may be obtained through simple queries.

Table 2

30

- Number of collected utterances;
- Percentage of rejected utterances for a given grammar;
- Average length of an utterance;
- Call volume in a give data range;
- 5      • Popularity of a given grammar or dialog state; and/or
- Transcription management (i.e. transcriber's productivity).

Further, in operation **356**, the utterances in the database are transmitted to a plurality of users utilizing a network. As such, transcriptions of the utterances in the database  
10      may be received from the users utilizing the network. Note operation **358**. As an option, the transcriptions of the utterances may be received from the users using a network browser.

Figure **4** illustrates a web-based interface **400** that may be used which interacts with  
15      the database to enable and coordinate the audio transcription effort. As shown, a speaker icon **402** is adapted for emitting a present utterance upon the selection thereof. Previous and next utterances may be queued up using selection icons **404**. Upon the utterance being emitted, a local or remote user may enter a string corresponding to the utterance in a string field **406**. Further, comments (re.  
20      transcriber's performance) may be entered regarding the transcription using a comment field **408**. Such comments may be stored for facilitating the tuning effort, as will soon become apparent.

As an option, the web-based interface **400** may include a hint pull down menu **410**.  
25      Such hint pull down menu **410** allows a user choose from a plurality of strings identified by the speech recognition process. This allows the transcriber to do a manual comparison between the utterance and the results of the speech recognition process. Comments regarding this analysis may also be entered in the comment field **408**.

The web-based interface **400** thus allows anyone with a web-browser and a network connection to contribute to the tuning effort. During use, the interface **400** is capable of playing collected sound files to the authenticated user, and allows them to type into the browser what they hear. Making the transcription task remote  
5 simplifies the task of obtaining quality transcriptions of location specific audio data (street names, city names, landmarks). The order in which the utterances are fed to the transcribers can be tweaked by a transcription administrator (e.g. to favor certain grammars, or more recently collected utterances). This allows for the transcribers work to be focused on the areas needed.

10 Similar to the speech recognition process of operation **304** of Figure **3**, the present interface **400** of Figure **4** and the transcription process contribute information for use during subsequent tuning. Table 3 illustrates various fields of information that may be associated with each utterance record in the database.

15 Table 3

- Date the utterance was transcribed;
- Identifier of the transcriber;
- Transcription text;
- Transcription comments noting speech anomalies;  
and/or
- Gender identifier.

25 During operation, the database of utterances collected and maintained during the methods of Figures **3** may be used to provide various services. Examples of various specific voice portal applications are set forth in Table 4. It should be noted that any services may be afforded per the desires of the user.

30

Table 4

- 5 • **Nationwide Business Finder** - search engine for locating businesses representing popular brands demanded by mobile consumers.
- **Nationwide Driving Directions** - point-to-point driving directions
- **Worldwide Flight Information** - up-to-the-minute flight information on major domestic and international carriers
- 10 • **Nationwide Traffic Updates** - real-time traffic information for metropolitan areas
- **Worldwide Weather** - updates and extended forecasts throughout the world
- 15 • **News** - audio feeds providing the latest national and world headlines, as well as regular updates for business, technology, finance, sports, health and entertainment news
- 20 • **Sports** - up-to-the-minute scores and highlights from the NFL, Major League Baseball, NHL, NBA, college football, basketball, hockey, tennis, auto racing, golf, soccer and boxing
- 25 • **Stock Quotes** - access to major indices and all stocks on the NYSE, NASDAQ, and AMEX exchanges
- **Infotainment** - updates on soap operas, television dramas, lottery numbers and horoscopes
- 30

Figure 5 is a schematic illustrating the manner in which VoiceXML functions in the context of the aforementioned architecture to support a voice portal that provides services such as those of Table 4. As shown, a typical VoiceXML interpreter 500 runs on a specialized voice gateway node 502 that is connected both to the public switched telephone network 504 and to the Internet 506. As shown, VoiceXML 508 acts as an interface between the voice gateway node 502 and the Internet 506.

VoiceXML takes advantage of several trends:

- The growth of the World-Wide Web and of its capabilities.
  - Improvements in computer-based speech recognition and text-to-speech synthesis.
  - The spread of the WWW beyond the desktop computer.
- Voice application development is easier because VoiceXML is a high-level, domain-specific markup language, and because voice applications can now be constructed with plentiful, inexpensive, and powerful web application development tools.

VoiceXML is based on XML. XML is a general and highly flexible representation of any type of data, and various transformation technologies make it easy to map one XML structure to another, or to map XML into other data formats.

VoiceXML is an extensible markup language (XML) for the creation of automated speech recognition (ASR) and interactive voice response (IVR) applications. Based on the XML tag/attribute format, the VoiceXML syntax involves enclosing instructions (items) within a tag structure in the following manner:

```
< element_name attribute_name="attribute_value">
.....contained items.....
< /element_name>
```



A VoiceXML application consists of one or more text files called documents. These document files are denoted by a ".vxml" file extension and contain the various VoiceXML instructions for the application. It is recommended that the first instruction in any document to be seen by the interpreter be the XML version tag:

5

```
< ?xml version="1.0"?>
```

The remainder of the document's instructions should be enclosed by the vxml tag with the version attribute set equal to the version of VoiceXML being used ("1.0" in the present case) as follows:

10

```
< vxml version="1.0">
```

Inside of the < vxml> tag, a document is broken up into discrete dialog elements.

15

Each element has a name and is responsible for executing some portion of the dialog. An element is denoted by the use of the < element > tag. Table 5 illustrates an exemplary list of element types available in one specification of VoiceXML.

20

#### Table 5

##### **element types:**

25

**<field>** - gathers input from the user via speech or DTMF recognition as defined by a grammar

**<record>** - records an audio clip from the user

30

**<transfer>** - transfers the user to another phone number

**<object>** - invokes a platform-specific object that may gather user input, returning the result as an ECMAScript object

`<subdialog>` - performs a call to another dialog or document(similar to a function call), returning the result as an ECMAScript object

- 5 Figure 6 illustrates a method 600 of dynamically extending element types for VoiceXML. Initially, in operation 602, a plurality of element types are registered with a VoiceXML interpreter. In one embodiment, the element types may be extended using JAVA. Of course, other computer languages may be used per the desires of the developer.

10

The registration process includes using a predetermined data structure to extend VoiceXML functionality in the VoiceXML interpreter. In one embodiment, such data structure may include a VoiceXML element type (i.e. element) to be extended, a name (i.e. type) for the VoiceXML element type to be extended, a class (i.e. classid) to be loaded for the VoiceXML element type to be extended, and a location (i.e. archive) of a file containing class files associated with the identified class. Table 6 summarizes definitions of the aforementioned element, type, classid, and archive.

15

Table 6

20

<u>element</u>	VoiceXML that developer wants to extend.
<u>type</u>	name of the new type being declared.
<u>classid</u>	fully-qualified name of the Java class to be loaded.
<u>archive</u>	A Jar archive containing the Java class files for classid and any other classes it requires.

25

- 30 The class referred to by classid in Table 6 extends the abstract class like the one in Table 7, and implements its abstract methods. Table 7 is an example for the case in which the element being extended is "field."

Table 7

35 package bevocal.vxml.extensions;

```
public abstract class FieldType {  
5     abstract FieldGrammar getGrammar(String type, Map params);  
        boolean validate(String result) { return true; }  
  
    protected static class FieldGrammar {  
        public FieldGrammar(String mimeType, String grammar);  
10        public final String getMimeType();  
        public final String getGrammarString();  
    };  
};
```

- 15 More information will now be set forth regarding the methods, `getGrammar` and `validate`, shown in Table 7. It is important to note that the foregoing is merely an example of extending a `<field>` element.

#### *GetGrammar*

- 20 The `getGrammar` method shown in Table 7 is called when the VoiceXML document containing the extended element type is parsed. It may return an `FieldGrammar` object containing a string representation of the grammar for the extended element type, along with the MIME type of the grammar. Additional grammar types may
- 25 also be supported. The type argument to `getGrammar` indicates the type of extended element being created, which allows one to use the same Java class to implement several different extended element types. The params argument is a map containing any parameters used on the element type. For example, if the type were
- 30 “duration?increment=15”, type would be “duration”, and params would contain the single key/value pair {“increment”, “15”}.

#### *Validate*

- The `validate` method shown in Table 7 is called after the interpreter has recognized
- 35 an utterance that matches the grammar returned by the `getGrammar` method, but before the element type variable is set and the `<filled>` blocks are executed. The present method performs post-processing to make sure that the value is within the

accepted range for this element; its argument is the string value that corresponds to the user's utterance. If the potential result is not valid, validate returns false, which causes a nomatch event to be issued.

- 5 Ideally, an element's grammar does not accept any utterances that aren't valid for the element. However, sometimes it is difficult to construct a grammar that can filter out all invalid responses, and the validate method provides one with an additional degree of flexibility in those cases. If one does not override validate, it will return true by default, meaning that all utterances that match the element's grammar are  
10 valid.

### *Namespaces*

- 15 In a preferred embodiment, the registration operation 602 further includes tagging the registered element types as being extensions to a conventional set of element types. In one embodiment, the element types may be tagged utilizing extensible mark-up language (XML) namespaces. This is to ensure that the tag name does not conflict with any tags that are added to VoiceXML in the future.

- 20 A namespace refers to a document at a specific Web site that identifies the names of particular data elements or attributes used within the XML file. The XML file creator identifies the namespace by specifying its Web address (URL) near the beginning of the XML file. A XML parser, usually provided as part of a Web browser, then knows where to find the rules for displaying and other information  
25 about each element in the XML file.

- In use, the registered element types are received during use of the VoiceXML interpreter. Note operation 604. Such registered element types are identified by the aforementioned tagging. In response to such receipt, in operation 606, code  
30 associated with the registered element types is accessed utilizing the VoiceXML interpreter. Such code serves to extend the functionality of the VoiceXML, as

indicated in operation 608. It should be noted that grammar extensions may also be employed per the desires of the user.

### *JAVA implementation*

5

As mentioned earlier, the code may be written in JAVA. JAVA solves many of the client-side problems by:

- Improving performance on the client side;
- Enabling the creation of dynamic, real-time Web applications; and
- 10 • Providing the ability to create a wide variety of user interface components.

With JAVA, developers can create robust User Interface (UI) components. Custom "widgets" (e.g., real-time stock tickers, animated icons, etc.) can be created, and client-side performance is improved. Unlike HTML, JAVA supports the notion of  
15 client-side validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

JAVA has emerged as an industry-recognized language for "programming the  
20 Internet." JAVA is defined as: a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. JAVA supports programming for the Internet in the form of platform-independent JAVA applets. JAVA applets are small, specialized applications that comply with the  
25 JAVA Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g., simple animations, page adornments, basic games, etc.). Applets execute within a JAVA -compatible browser (e.g., Netscape Navigator) by copying code from the server to client. From a language standpoint, JAVA's core feature set is based on C++.

30

### *Extending Type Attributes*

Many pre-existing VoiceXML elements such as <field> and <grammar> have a “type” attribute that controls the workings of such elements. For example, with respect to field, the type can be “boolean” or “date”, which controls whether the field accepts a “yes”/“no” response, or a date.

5

The present extension provides a way to extend the set of “type” attributes that a pre-defined element such as <field> accepts. One can register the tag name (i.e. “field”, “grammar”, etc.), the extended type attribute name (i.e. “country”, etc.), and the class to be loaded to implement that extended type attribute. Later, when the interpreter encounters the tag (i.e. <field type=“country”>), it may use the mapping to determine which code to use to implement the type attribute.

10

A system, method and computer program product are thus provided for dynamically extending a type attribute of elements of a voice-based extensible mark-up language (VoiceXML). Initially, an extended type attribute associated with an element of VoiceXML is registered with a VoiceXML interpreter. During use, the element may be received, and the extended type attribute associated with the element is identified. Thereafter, code corresponding to the registered type attribute may be accessed utilizing the VoiceXML interpreter. Such code extends the functionality of the VoiceXML.

15

20

As such, a data structure is provided for dynamically extending a type attribute of elements of a VoiceXML. First provided is a VoiceXML type attribute object that is extended to include a previously undefined type attribute. Also included is a VoiceXML element. Associated therewith is a class object for identifying a class to be loaded for the VoiceXML type attribute object that is extended. In use, the data structure is capable of being used to register type attributes capable of accessing code to extend the functionality of the VoiceXML.

25

30

*Example*

An example of the foregoing extension technique will now be set forth. Table 8 illustrates an exemplary data structure for registering an extended type attribute, “duration,” associated with the “field” element. This extension may be particular useful since it extends the basic types of what users can say.

5

Table 8

10                   <bevocal:extend element="field" type="duration"  
                    classid="com.foo.vxml.mydate"  
                    archive="extensions.jar" />

15                   Since such a type attribute is not supported by default, developers are thus provided  
                    with a way of adding the same. An example of how such extended field type  
                    attribute is added is shown in Table 9.

Table 9

20                   <field type="duration" name="length">  
                    How long an appointment do you need?  
                    ...  
                  </field>

25                   It should be noted that various other type attributes may be supported including, but  
                    not limited to digits, number, phone, currency, equity, airline information, address,  
                    country, or any other functionality required by a voice portal.

30                   While various embodiments have been described above, it should be understood that  
                    they have been presented by way of example only, and not limitation. Thus, the  
                    breadth and scope of a preferred embodiment should not be limited by any of the  
                    above-described exemplary embodiments, but should be defined only in accordance  
                    with the following claims and their equivalents.